



Create curved borders with CSS

Curved box edges are a smart, but fiddly, design touch. Matt Preston shows you a couple of ways to get them right

Using CSS to design websites has two main advantages: separating the design from the code means that you can easily update the look and feel of your website, and it simplifies the design, too. Without CSS, the only way to draw a box with curved edges is to use a table and create images that look like curved edges in the top-left, top-right, bottom-left and bottom-right cells. It works, but is a lot of code, hard to read and difficult to update. CSS makes it easier by separating the content, stored in the HTML page, from its presentation.

CSS is best stored in a separate file, which you can then reuse throughout your site. This also means that editing one CSS file automatically updates your entire site's design. These files are saved with a .css extension and are linked to HTML pages with the line `<link rel="stylesheet" type="text/css" href="test.css">` in the `<head>` section. CSS can also be used directly in the HTML page between the `<style>` and `</style>` tags written in the `<head>` section. We've used this method in the example on the cover disc and at www.shopperexpert.co.uk, so you can see clearly what's going on.

In either case, the CSS is a list of rules telling the web browser how the content should look. Each rule consists of a selector, defining which HTML tags the rule applies to, and properties describing how the tag should be displayed. There are three types of rule, so each of the following creates a paragraph in a red font:

```
p {color: red;}
.content {color: red;}
#title {color: red;}
```

The first entry states that any paragraph should be written in red. The second two are, respectively, a Class (names start with a '.') and an ID (names start with a '#'). Both are similar, but an ID is a unique identifier that should be used only once on a page, while a Class can be used many times. Both are used in a similar way, as an attribute to any HTML tag. For example, the HTML `<p class="content">` and `<p id="title">` would both create a paragraph with red text (we don't need the leading '.' or '#'), but we could apply them to any other tag, such as `<ul>` for an unordered list. As you

can see, adjusting the color tag in the CSS style sheet for any of these examples would automatically update our website's design. For more information, see www.w3schools.com/css, which has an excellent tutorial and reference for CSS.

LEARNING CURVE

The first method of drawing curved boxes in CSS requires us to define our style sheet to include the necessary images for the corners. Instead of using a table for the layout though, we'll use the `<div>` tag; this defines a section of a webpage – in our case, a box.

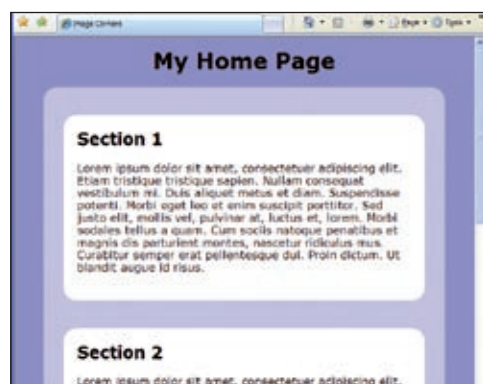
Before we start, we need to create the curved image files, with each curve the same colour as the box we want to create, and having the same background colour as our webpage. Most image editors will let you create curved boxes from which you can copy and paste the corners, but Google's online tool (<http://tinyurl.com/62nevk>) is easier to use. It will draw a 20-pixel-high, 20-pixel-wide red corner with a white background that's suitable for use as the top-left corner of a box.

You can easily change parameters. Altering the hexadecimal number after 'c=' changes the colour of the curve, while altering the number after 'bc=' changes the background colour (see www.december.com/html/spec/color.html for a list of web colours). The height and width in pixels are the numbers after 'w=' and 'h='. To change the type of corner change 'a=' to tl, tr, br and bl. While you could just use these links to directly draw your corners, we don't recommend it as Google could remove the service at any point. Instead, create all four corners and save them.

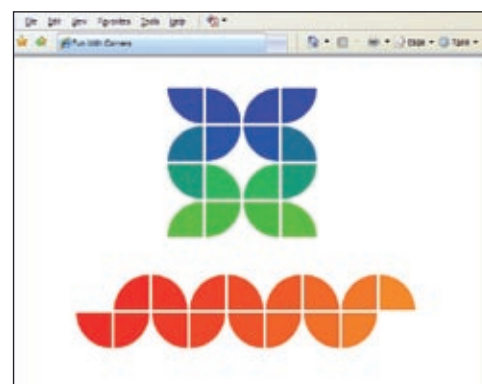
Our example, `cornerimages.html`, creates a red box using the `top-left.png`, `top-right.png`, `bottom-left.png` and `bottom-right.png` files. These are on the cover disc.

First, our HTML file defines the CSS:

```
.round-box {
    width: 250px;
    background-color: red;
}
.round-top {
    background: url(top-right.png) no-repeat
    top right;
```



▲ Using CSS allows you to create a set style for boxes that can be applied time after time



▲ Google's online tool for corner creation allows you to pre-prepare your designs easily

INTRODUCTION

A neat way to improve your website's design is to add curved edges to any boxes you might have. Of course, this sounds easier than it actually is, as there's no standard way to add curves in HTML. Fortunately, with Cascading Style Sheets (CSS), which separate the design of a webpage from the HTML code, there are several tips, tricks and hacks to get the required results. There are several different methods, each with its own advantages and disadvantages. We'll take you through each method, and you can choose the one you prefer in your own designs.

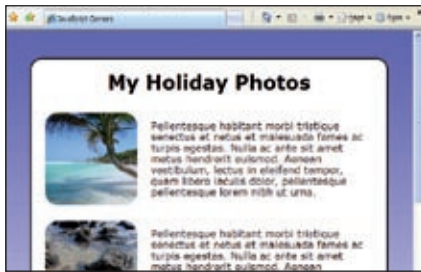
You can see each of these examples online at www.shopperexpert.co.uk, or look at the HTML files directly on the cover disc. Dedicated web hosting is not required for this project.

Matt Preston

Web developer



matt@computershopper.co.uk



▲ Getting your designs from a JavaScript library also offers an easy option

```

}
.round-bottom {
  background: url(bottom-right.png)
  no-repeat top right;
}
.corner {
  width: 20px;
  height: 20px;
  border: none;
  display: block;
}

```

We can only define the top-right and bottom-right images in our CSS classes, and we add the extra images and <div> sections in the HTML:

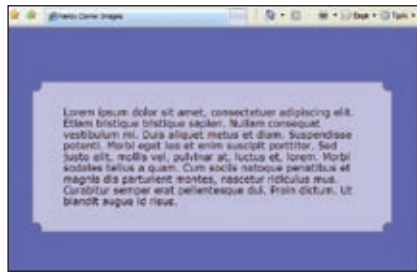
```

<div class="round-box">
  <div class="round-top">
    
  </div>
  <p>content goes here</p>
<div class="round-bottom">
  
</div>
</div>

```

This code creates a box with the properties and colours defined in the round-box Class. Inside this it creates another <div> section that uses the round-top Class to display the top-right image. Inside this box we use a standard HTML img tag to display the top-left image using the attributes in the corner Class. The content goes next and appears inside the main box.

We finish in a similar way to the top of the box with another box for the bottom-right and bottom-left images. This method works well and will work with most web browsers. You can put boxes in boxes and create fancier corners, too.



▲ However, creating discrete images for corners lets you be more adventurous with your designs

The boxinbox.html and fancycorners.html files are available on the cover CD and online to show what can be achieved.

AUTOMATIC CORNERS

While drawing the images certainly works, you can do the job automatically using the excellent curvyCorners JavaScript library available for free from www.curvycorners.net. You'll need to extract the rounded_corners_lite.inc.js file. Our curvycorners.html file shows this working.

First, the JavaScript library needs to be included in the <head> section of the HTML document with the line <script type="text/javascript" src="rounded_corners_lite.inc.js"></script> (you can change the path to this file if you want to store it somewhere else).

Next, we define the rounded-box Class in the <style> section.

```

.rounded-box {
  width: 250px;
  background-color: red;
  padding: 10px;
}

```

This is standard CSS for defining an element 250 pixels wide with a red background, but we also need to define the JavaScript that will draw the curved lines automatically:

```

<script type="text/JavaScript">
  window.onload = function()
  {
    var _settings = {
      tl: { radius: 20 },
      tr: { radius: 20 },
      bl: { radius: 20 },
      br: { radius: 20 },
      antiAlias: true,
      autoPad: true
    }

```

```

}
var _myBox = new curvyCorners(
  _settings, "rounded-box");
_myBox.applyCornersToAll();
}
</script>

```

First, this creates a variable called '_settings' that we use to store the settings including the radius of each corner in pixels. Setting antiAlias to true makes the curves look smoother, although they take longer to draw. Finally the variable '_myBox' is an object that draws curves according to the parameters in the '_settings' variable and with the rounded-box Class. The final line tells the script to draw a curved box every time it sees the rounded-box Class.

Drawing a curved box is now easy. In the <body> part of the HTML document we can use:

```

<div class="rounded-box">
  <p>Content goes here.</p>
</div>

```

The beauty of this method is that we can expand it to draw curved boxes around other Classes. To do this we have to create a new variable and give it the same settings, but a different Class name:

```

var _yournewvariable = new curvyCorners(_settings, "yourclassname");

```

This can be applied to all matching Classes with:

```

_yournewvariable.applyCornersToAll();

```

You can see this in action with our javacorners.html file, which draws curved boxes around the table entries that use the round-image Class.

Both the methods here have their strengths, but in most cases it is easier to use JavaScript as it doesn't require you to make corner images first. We'd suggest only using the image method if you want more than basic curves. Check our examples online and on the cover disc to see how far you can take your designs.

Next month

GOOGLE MAPS

Find out how to add Google Maps to your website

NOBODY HOSTS MORE WEBSITES IN THE UK THAN US. FACT!

*Source: Netcraft

WEB HOSTING

FROM
£5.95
PER MONTH

VPS

FROM
£14.99
PER MONTH

DEDICATED SERVERS

FROM
£59.99
PER MONTH

RESELLER PACKAGES

FROM
£26.99
PER MONTH

WebFusion
WHERE WEBHOSTING WORKS